

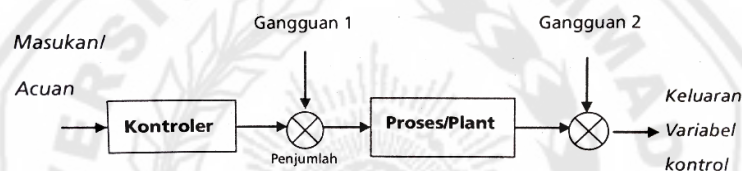
BAB II

TINJAUAN PUSTAKA

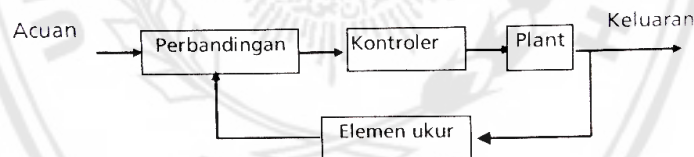
2.1 SISTEM KONTROL

(Hakim, 2012) Sistem kontrol merupakan kombinasi dari beberapa komponen yang bekerja bersama-sama secara timbal balik dan membentuk konfigurasi sistem yang memberikan suatu hasil yang dikehendaki. Hasil ini sering dinamakan sebagai tanggapan sistem (*system response*).

Sistem kontrol dibagi menjadi dua yaitu: sistem kontrol lup terbuka (*open loop control system*) dan sistem kontrol lup tertutup (*closed loop control system*). Sistem kontrol lup terbuka merupakan sistem kontrol yang hasil keluarannya tidak mempunyai pengaruh terhadap aksi kontrol, sedangkan sistem kontrol lup tertutup disebut sistem kontrol umpan balik (*feedback*) adalah sistem kontrol yang cenderung memperkecil selisih keluaran sistem dan acuan masukan (Hakim, 2012:4).

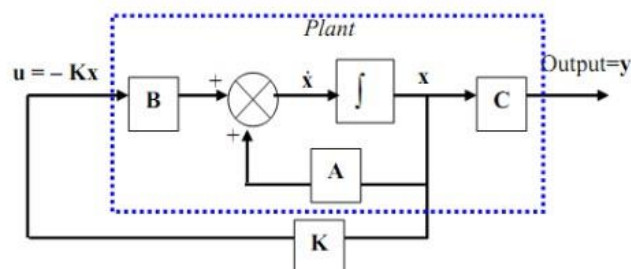


Gambar 2.1 Sistem kontrol lup terbuka (Hakim, 2012:3)



Gambar 2.2 Sistem kontrol umpan balik

2.2 KONSEP SISTEM KONTROL OPTIMAL METODE LINEAR QUADRATIC REGULATOR



Gambar 2.3 Diagram blok LQR

(Bofy Panji Prayudha, 2017) Sistem optimal adalah sistem yang mempunyai unjuk kerja terbaik (*best performance*) terhadap suatu acuan tertentu. Sistem kontrol optimal memerlukan adanya suatu kriteria optimasi yang dapat meminimumkan hasil pengukuran dengan deviasi perilaku sistem terhadap perilaku idealnya.

Pengukuran tersebut dilakukan dengan menentukan indeks performansi, yang merupakan suatu fungsi dari suatu harga yang dapat dianggap menunjukkan seberapa besar kinerja sistem yang sesungguhnya sesuai dengan kinerja yang diinginkan.

$$J = \int_{t_i}^{t_f} L(x, u, t) dt \text{ bila } J \text{ minimum, maka sistem optimal} \quad (2.1)$$

Supaya sistem tersebut dapat dikontrol, maka perlu dibuat model matematis yang menghubungkan antar masukan(input) dan keluaran(output). Pada sistem kontrol optimal model yang banyak digunakan adalah persamaan keadaan. Dalam persamaan keadaan persamaan diferensial berorde satu secara simultan, dan ditulis dalam notasi vektor matriks:

$$[X]' = [A][X] + [B][U] \quad (2.2)$$

$$[Y] = [C][X] + [D][U] \quad (2.3)$$

2.2.1 Teori Regulator Optimal

Linear Quadratic Regulator merupakan salah satu metode dalam perancangan sistem kontrol optimal. Permasalahan dapat dirumuskan dan dipecahkan pada kawasan frekuensi menggunakan fungsi alih.

$$J = \int_0^{\infty} e^2 dt \quad (2.4)$$

$$J = \frac{1}{2} \int_0^{\infty} [X^T Q X + U^T R U] dt \quad (2.5)$$

Q= matriks simetris, semi definit positif, real ($Q > 0$)

R= matriks simetris, definit positif, real ($R > 0$)

Permasalahannya adalah bagaimana meminimumkan suatu *cost function* J. hal ini dikenal dengan permasalahan optimasi sistem dengan metode *Linear Quadratic Regulator*(LQR). Jika sistem tersebut *scalar*, maka *cost function* menjadi persamaan:

$$J = \frac{1}{2} \int_0^{\infty} [X^T Q X + U^T R U] dt \quad (2.6)$$

J merupakan representasi dari jumlah energy dan sinyal kontrol.

2.2.2 Linear Quadratic Regulator(LQR)

Metode optimasi dengan *Linear Quadratic Regulator(LQR)* adalah dengan menentukan sinyal masukan yang akan memindahkan suatu state sistem linier dari kondisi awal $x(t_0)$ menuju ke suatu kondisi akhir $x(t)$ yang meminimumkan suatu indeks unjuk kerja performansi kuadratis. *Cost functional* yang dimaksud adalah waktu integral dari bentuk kuadratis pada vektor keadaan (state) x dan vektor masukan u seperti pada persamaan:

$$X^T Q X + U^T R U \quad (2.7)$$

Dimana Q adalah matriks semi definit positif dan R adalah matriks definit positif. Dengan dasar seperti diatas, variasi parameter dari masalah perancangan *Linear Quadratic Regulator(LQR)* dapat ditentukan juga untuk kondisi akhir, yang mungkin dapat berpengaruh pada *cost function*.

2.2.3 Kontroller Algebraic Ricatti Equations (CARE)

Untuk sistem linier, time-invariant, dapat diturunkan persamaan Aljabar Ricatti untuk mencari solusi optimal sebagai berikut:

$$J = \frac{1}{2} \int_0^{\infty} [X^T Q X + U^T R U] dt \quad (2.8)$$

$$H(x, \lambda, t) = \frac{1}{2} [X^T Q X + U^T R U] + \lambda^T f(x, u, \lambda, t) \quad (2.9)$$

$$H(x, \lambda, t) = \frac{1}{2} [X^T Q X + U^T R U] + \lambda^T (Ax + Bu) \quad (2.10)$$

$$\text{Kondisi optimal : } \frac{\partial H}{\partial U} = 0$$

$$\text{Persamaan state : } \dot{x} = \frac{\partial H}{\partial \lambda}$$

$$\text{Persamaan costate : } -\lambda' = \frac{\partial H}{\partial x}$$

Dengan menggunakan aturan diferensial matriks dan vektor, persamaan berikut menjadi:

$$\dot{x} = Ax + Bu \quad (2.11)$$

$$\dot{\lambda} = Qx + A^T \lambda \quad (2.12)$$

$$u' = -R^{-1}B^T \lambda \text{ dengan } u' \text{ merupakan vektor kontrol optimal.} \quad (2.13)$$

Dengan mensubstitusikan persamaan kontrol optimal kedalam persamaan state, kita dapatkan:

$$\begin{bmatrix} \dot{x}' \\ \dot{y}' \end{bmatrix} = \begin{bmatrix} A & -BR^{-1} \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = H \begin{bmatrix} x \\ \lambda \end{bmatrix} \quad (2.14)$$

H disebut matriks Hamiltonian dan sangat berperan penting dalam teori LQR. Dengan menggunakan substitusi $\lambda = Px$, kemudian dilakukan diferensiasi pada kedua ruas, diperoleh:

$$\frac{d\lambda}{dt} = \frac{dP}{dt}x + P \frac{dx}{dt} = \frac{dP}{dt}x + PAx - PBR^{-1}B^TPx = -Qx - A^TPx \quad (2.15)$$

Persamaan harus dapat memenuhi semua nilai x, syarat cukup untuk kontrol optimal matriks P harus memenuhi:

$$\frac{dP}{dt} = A^TP + PA + Q - PBR^{-1}B^TP = 0 \quad (2.16)$$

Persamaan diatas dikenal sebagai persamaan Ricatti(*Ricatti Equation*).

$$u(t) = -[k].x(t) \quad (2.17)$$

$$k(t) = R^{-1}B^TP \quad (2.18)$$

Dengan syarat matriks A dan B, *kontrollable* dan *observable*.

Dengan adanya umpan balik K, persamaan keadaan akan berubah menjadi persamaan:

$$\dot{x} = (A - BK) x \quad (2.19)$$

2.3 METODE RUNGE-KUTTA

(Iffatul Mardhiyah, 2007) Penyelesaian PDB dengan metode deret Taylor tidak praktis karena metode tersebut membutuhkan perhitungan turunan $f(x, y)$. Lagipula, tidak semua fungsi mudah dihitung turunannya, terutama bagi fungsi yang bentuknya rumit. Semakin tinggi orde metode deret Taylor, semakin tinggi turunan fungsi yang harus dihitung. Karena pertimbangan ini, metode deret Taylor yang berorde tinggi pun tidak dapat diterima dalam masalah praktek. Bentuk umum metoda Range-Kutta orde- n ialah:

$$y_{r+1} = y_r + a_1 k_1 + a_2 k_2 + \dots + a_n k_n \quad (2.20)$$

dengan a_1, a_2, \dots, a_n adalah tetapan, dan

$$k_1 = hf(x_r, y_r) \quad (2.21)$$

$$k_2 = hf(x_r + p_1 h, y_r + q_{11} k_1) \quad (2.22)$$

$$k_3 = hf(x_r + p_2 h, y_r + q_{21} k_1 + q_{22} k_2) \quad (2.23)$$

...

$$k_n = hf(x_r + p_{n-1} h, y_r + q_{n-1,1} k_1 + q_{n-1,2} k_2 + \dots + q_{n-1,n-1} k_{n-1}) \quad (2.24)$$

Nilai a_i, p_i, q_{ij} dipilih sedemikian rupa sehingga meminimumkan galat per langkah, dan persamaan (P.8.24) akan sama dengan metode deret Taylor dari orde setinggi mungkin..

Galat per langkah metode Runge-Kutta orde- $n : O(h^{n+1})$

Galat longgokan metode Runge-Kutta orde- $n : O(h^n)$

Orde metode = n

2.3.1 Metode Runge-Kutta Orde Dua

Metode Runge-Kutta orde dua berbentuk

$$k_1 = hf(x_r, y_r) \quad (2.25)$$

$$k_2 = hf(x_r + p_1 h, y_r + q_{11} k_1) \quad (2.26)$$

$$y_{r+1} = y_r + (a_1 k_1 + a_2 k_2) \quad (2.27)$$

Galat per langkah metode Runge-Kutta orde dua adalah $O(h^3)$.

Galat longgokan metode Runge-Kutta orde dua adalah $O(h^2)$.

Nilai a_1, a_2, p_1 , dan q_{11} ditentukan sebagai berikut:

Misalkan

$$f_r = f(x_r, y_r) \\ f_x = \frac{\partial f(x_r, y_r)}{\partial x}, \text{ dan } f_y = \frac{\partial f(x_r, y_r)}{\partial y} \quad (2.28)$$

Uraikan k_2 ke dalam deret Taylor di sekitar (x_r, y_r) sampai suku orde satu saja:

$$\begin{aligned}
 k_2 &= hf(x_r + p_1 h, y_r + q_1 k_1) \\
 &= h(f + p_1 h f_x + q_1 k_1 f_y) \\
 &= h(f + p_1 h f_x + q_1 h f f_y) \\
 &= h(f + h(p_1 f_x + q_1 f f_y))
 \end{aligned} \tag{2.29}$$

Sedangkan k_1 tidak perlu diuraikan karena sudah berada dalam bentuk (x_r, y_r) .

Jadi,

$$\begin{aligned}
 y_{r+1} &= y_r + a_1 k_1 + a_2 k_2 \\
 &= y_r + a_1 h f_r + a_2 h f + a_2 h^2 (p_1 f_x + q_1 f f_y) \\
 &= y_r + (a_1 + a_2) h f + a_2 h^2 (p_1 f_x + q_1 f f_y)
 \end{aligned} \tag{2.30}$$

Uraikan Y_{r+1} sejati di sekitar x_r sampai suku orde dua saja:

$$Y_{r+1} = y_r + h y_r' + \frac{1}{2} h^2 y_r'' \tag{2.31}$$

Mengingat

$$y_r' = f(x_r, y_r) = f_r \tag{2.32}$$

dan

$$\begin{aligned}
 y_r'' &= f'(x_r, y_r) = \\
 &= \frac{df(x_r, y_r)}{dx} \\
 &= \frac{\partial f}{\partial x} \frac{dx}{dx} + \frac{\partial f}{\partial y} \frac{dy}{dx} \\
 &= f_x + f_y f_r \\
 &= f_x + f f_y
 \end{aligned} \tag{2.33}$$

maka persamaan menjadi:

$$y_{r+1} = y_r + h f + \frac{1}{2} h^2 (f_x + f f_y) \tag{2.34}$$

Galat per langkah metode adalah

$$E_p = \{ h f + \frac{1}{2} h^2 (f_x + f f_y) \} - \{ (a_1 + a_2) h f + a_2 h^2 (p_1 f_x + q_1 f f_y) \}$$

Dengan membuat galat per langkah $E_p = 0$,

$$0 = \{ h f_r + \frac{1}{2} h^2 (f_x + f_r f_y) \} - \{ (a_1 + a_2) h f_r + a_2 h^2 (p_1 f_x + q_1 f_r f_y) \}$$

atau

$$h f_r + \frac{1}{2} h^2 (f_x + f_r f_y) = (a_1 + a_2) h f_r + a_2 h^2 (p_1 f_x + q_1 f_r f_y) \tag{2.35}$$

Agar ruas kiri dan ruas kanannya sama, haruslah

$$a_1 + a_2 = 1$$

$$a_2 p_1 = \frac{1}{2}$$

$$a_2 q_{11} = \frac{1}{2}$$

Karena sistem persamaan di atas terdiri atas tiga persamaan dengan empat peubah yang tidak diketahui, maka solusinya tidak unik, Solusi yang unik hanya dapat diperoleh dengan memberikan sebuah peubah dengan sebuah harga. Misalkan ditentukan nilai $a_2 = t$, $t \in \mathbb{R}$, maka

$$a_1 = 1 - a_2 = 1 - t$$

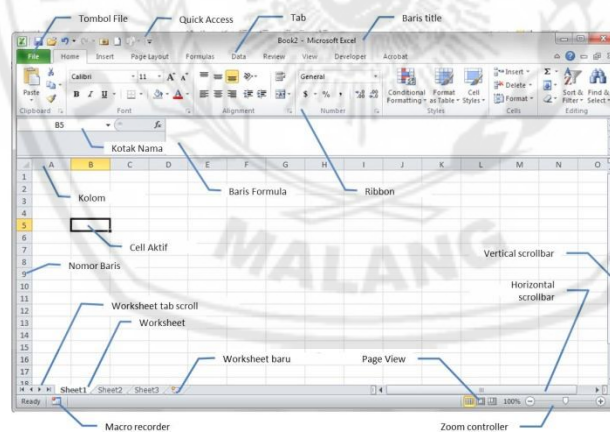
$$p_1 = \frac{1}{2a_2} = \frac{1}{2t}$$

$$q_{11} = \frac{1}{2a_2} = \frac{1}{2t}$$

(2.36)

2.4 MICROSOFT EXCEL

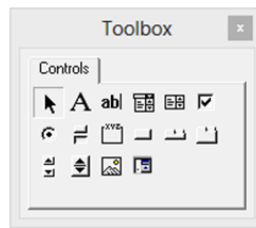
Microsoft Excel atau *Microsoft Office Excel* adalah sebuah program aplikasi lembar kerja spreadsheet yang dibuat dan didistribusikan oleh Microsoft Corporation yang dapat dijalankan pada Microsoft Windows dan Mac OS. *Microsoft Excel* memiliki beberapa kelebihan dibandingkan *software* lainnya, seperti olah angka dan grafik, adanya rumus-rumus logika, adanya bahasa pemrograman, mudah diprogram dan ketersediaannya yang luas di setiap komputer(I Kadek Sembah Semadiartha,2012)



Gambar 2.4 Lembar kerja *Microsoft Excel*

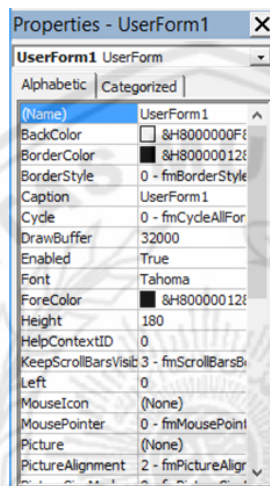
2.4.1 *Microsoft Visual Basic for Application (VBA)*

(Akhmad Arifin & Harto Nuroso ,2011) *Microsoft Excel* memiliki kemampuan pemrograman melalui bahasa makronya yang disusun melalui Visual Basic for Application (VBA). Dengan bahasa makro *Microsoft Excel* menyediakan fasilitas bagi pengguna untuk menyusun program-program aplikasi yang lebih kompleks seperti layaknya menyusun program komputer dengan menggunakan bahasa-bahasa pemrograman.VBA



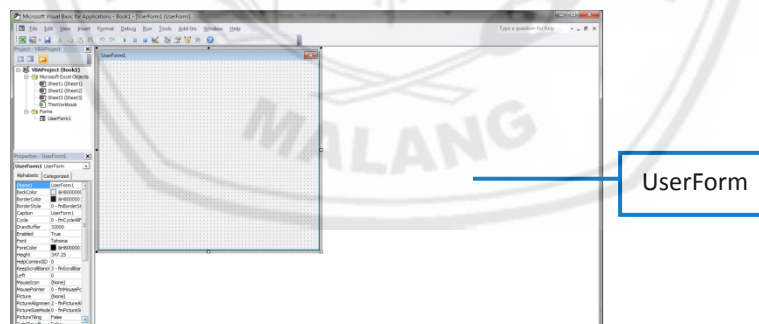
Gambar 2.5 *Toolboox control VBA*

ToolBox Control merupakan Objek dalam *Userform* atau *worksheet* yang dapat dimanipulasi, seperti *command button*, *text box*, *check box*, *combo box*, *list box*, *label*, dan *option button*.



Gambar 2.6 *Properties VBA*

Property, merupakan Karakteristik suatu objek seperti *ScrollArea*, *Font*, dan *Name*.



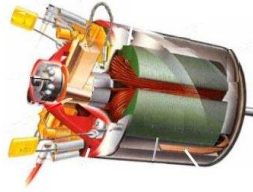
Gambar 2.7 *UserForm VBA*

UserForm merupakan Lembar kerja yang berisi kontrol dan instruksi VBA untuk memanipulasi antarmuka pengguna (*user interface*).

2.5 MOTOR DC

Motor DC untuk diaplikasikan pada kendali menggunakan gambar rangkaian ekuivalen untuk merepresentasikan motor DC dengan megnet permanen. Untuk analisis linier diasumsikan bahwa torsi yang dihasilkan motor sebanding dengan fluks pemisah

udara dan arus jangkar. Medan konstan, fluks konstan dan torsi mempunyai arah sesuai kumparan magnet, sehingga (Subrahmany, 1994):



Gambar 2.8 Motor DC

$$T_m(t) = K_T \cdot I_a \quad (2.37)$$

Karena konstan, maka persamaan (1) dapat ditulis

$$T_m(t) = K_T \cdot I_a \quad (2.38)$$

Tegangan keluaran *loop* terbuka dari persamaan (2) maka:

$$e_f(t) = K_T \quad (2.39)$$

Tegangan pada terminal jangkar motor diberikan oleh persamaan:

$$e_a = K_b \omega \quad (2.40)$$

Sehingga dengan menggunakan *Hukum Khirchoff* tegangan diperoleh:

$$e_a = I_a R_a + L_a \frac{di_a}{dt} + e_b \quad (2.41)$$

Ketika konduktor bergerak pada medan magnet, suatu tegangan dibangkitkan melintasi ujung-ujungnya. Tegangan ini, emf balik yang sebanding dengan kecepatan motor, berlawanan dengan aliran arus. Hubungan emf balik (e_b) Volt dengan kecepatan motor (ω_m) rad/detik adalah.

$$e_b = K_b \omega_m(t) \quad (2.42)$$

Persamaan Motor mempunyai keseimbangan torsi sebagai berikut:

$$J \frac{d\omega}{dt} + T_L = K_t I_a \quad (2.43)$$

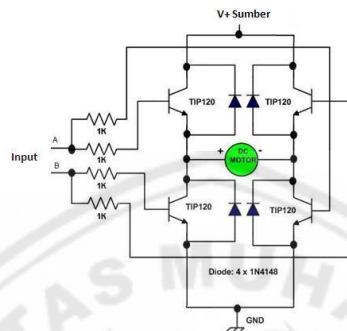
Dengan menggunakan rumus penguatan *Masson* (*Masson Gain Formula*):

$$T(s) = \frac{\frac{K_T}{J L_a}}{s^2 + s(L_a T_l + J R_a) + (R_a T_l + K_b K_t)} \quad (2.44)$$

Dimana e_a = Tegangan Masukan (Volt), K_t = Konstanta Torsi (Nm/A), J = Momen Inersia Ekvivalen (Kg m^2), L_a = Induktansi kumparan jangkar (H), T_L = Torka Beban (Nm), ω_m = Kecepatan motor (RPM), R_a = Tahanan jangkar (Ω), K_b = Konstanta EMF balik (V/s.rad), L_a = Induktansi kumparan jangkar (H), I_a = Arus kumparan jangkar (A).

2.6 DRIVER MOTOR DC H-BRIDGE TRANSISTOR

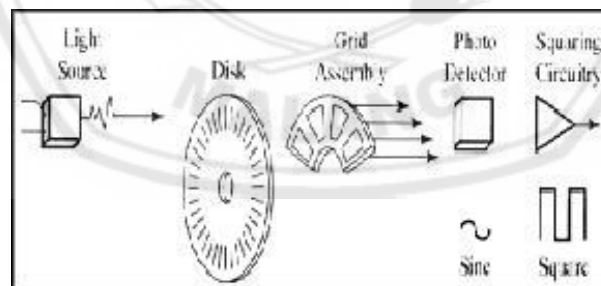
Driver motor DC tipe H-Bridge menggunakan power driver berupa transistor. Rangkaian driver motor DC H-Bridge transistor ini dapat mengendalikan arah putaran motor DC dalam 2 arah dan dapat dikontrol dengan metode PWM (pulse Width Modulation) maupun metode sinyal logika dasar TTL (High) dan (Low). Apabila menggunakan metode logika TTL 0 dan 1 maka rangkaian ini hanya dapat mengendalikan arah putaran motor DC saja dengan kecepatan putaran motor DC maksimum.



Gambar 2.9 *Driver Motor DC H-Bridge*

2.7 ROTARY ENCODER

Rotary encoder adalah divais elektromekanik yang dapat memonitor gerakan dan posisi. Rotary encoder umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh rotary encoder untuk diteruskan oleh rangkaian kendali.



Gambar 2.10 Blok penyusun Rotary Encoder

2.8 FPGA MyRio

2.8.1 NI MyRio-1900

MyRio adalah sebuah perangkat keras dimana pengguna dapat memanipulasi fungsi-fungsi untuk membuat berbagai sistem. MyRio menggunakan ARM Processor yang juga sebuah FPGA processor. Dengan menggunakan Fitur-fitur yang ada dapat dibuat sistem yang lebih kompleks. MyRio menggunakan LabView sebagai IDE.

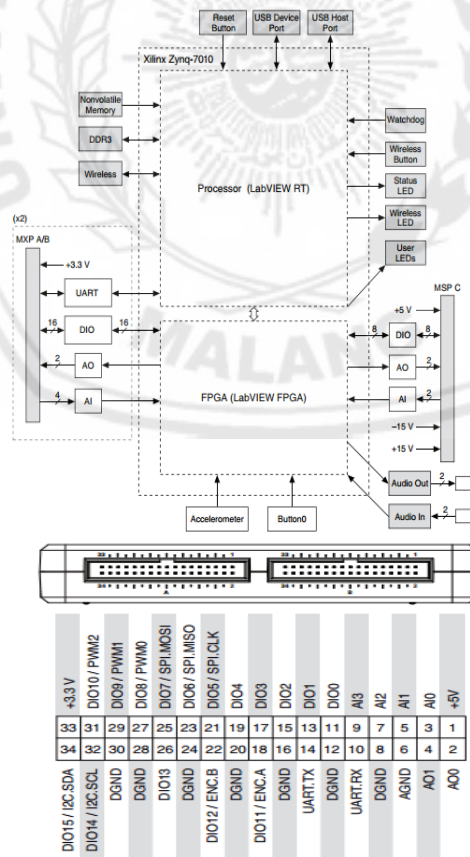


Gambar 2.11 NI MyRio-1900

Pada perangkat MyRio didukung oleh perangkat lunak yang berfungsi untuk pemrograman, meliputi :

- LabView
- LabView Real-Time Modul
- LabView myrio Toolkit

Berikut adalah diagram Blok dari NI-myRIO-1900 Hardware



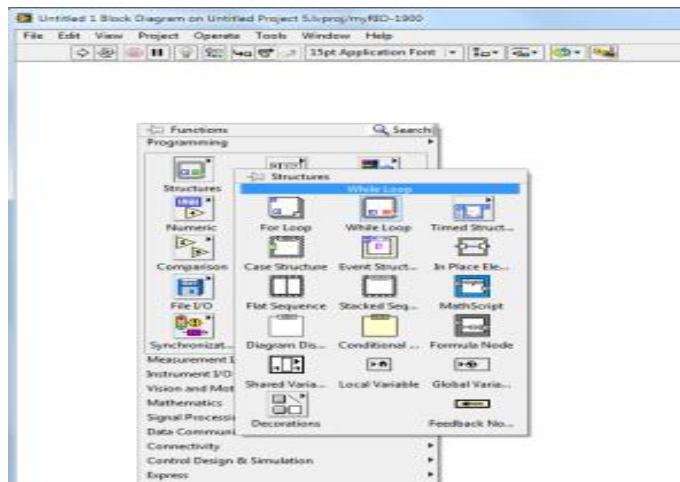
Gambar 2.12 Primary/Secondary Signals on MXP Connectors A dan B

Tabel 2.1 deskripsi sinyal pada MXP connectors A dan B

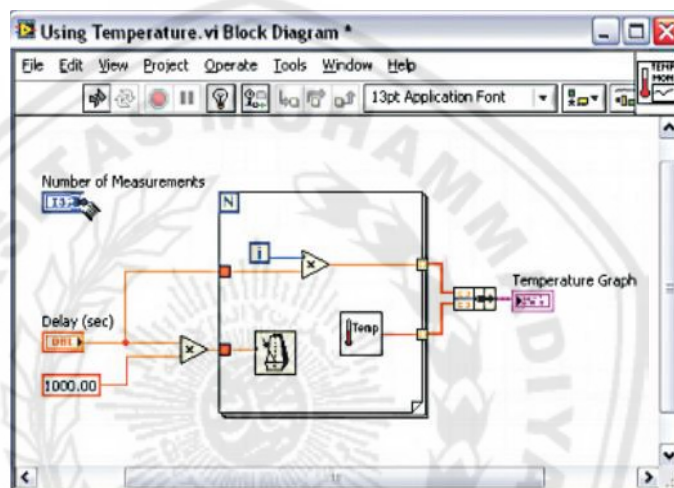
Signal Name	Reference	Direction	Description
+5 V	DGND	Output	+5 V power output.
AI <0..3>	AGND	Input	0-5 V, referenced, single-ended analog input channels, Refer to the <i>Analog Input Channels</i> section for more information.
AO <0..1>	AGND	Output	0-5 V, referenced, single-ended analog output channels, Refer to the <i>Analog Output Channels</i> section for more information.
AGND	N/A	N/A	Reference for analog input and output
+3.3 V	DGNG	Output	+3.3 V power output
DIO <0..5>	DGNG	Input or Output	General-purpose digital lines with 3.3 V output, 3.3 V/5 V-compatible input. Refer to the <i>DIO Line</i> section for more information.
UART.RX	DGND	Input	UART receive input. UART line are electrically identical to DIO lines.
UART.RX	DGND	Output	UART receive output. UART line are electrically identical to DIO lines.
DGND	N/A	N/A	Reference for digital signals, +5 V, and +3.3 V.

2.8.2 NI LabView

(Muhammad Sulaiman, 2016) Program LabVIEW disebut instrumen virtual, atau Vis, karena penampilan dan operasi mereka meniru instrumen fisik, seperti osiloskop dan multimeter. LabVIEW berisi seperangkat alat untuk memperoleh, menganalisis, menampilkan, dan menyimpan data. Di LabVIEW dapat membangun antarmuka pengguna, atau panel depan, dengan kontrol dan indikator.



Gambar 2.19 Tool pada Labview



Gambar 2.13 Pemrograman pada LabView

2.9 IDENTIFIKASI SISTEM

Dalam kerangka kesalahan prediksi, terdapat dua jenis metode identifikasi sistem yaitu metode parametrik dan *non*-parametrik. Metode parametrik ditandai oleh sejumlah parameter, sedangkan metode *non*-parametrik ditandai dengan model yang dihasilkan seperti kurva atau fungsi (Soderstrom dan Stoica, 1989). Contoh dari metode *non*-parametrik adalah identifikasi sistem *frequency domain*. Metode parametrik dapat diklasifikasikan menjadi *non*-rekursif dan rekursif.

Non-Rekursif Estimasi Parameter

Proses identifikasi *non*-rekursif terdiri dari estimasi parameter yang tidak diketahui dari dinamika sistem. Metode identifikasi berdasarkan algoritma *least-square* telah direkomendasikan untuk proses identifikasi untuk kemudahan implementasi dalam real system. Untuk proses identifikasi linear, model *discrete time* untuk sistem motor DC bisa digunakan. (Ljung, 2014)